Computer Olympic 2022

ตัวแปรชนิดโครงสร้าง (Structure Type)

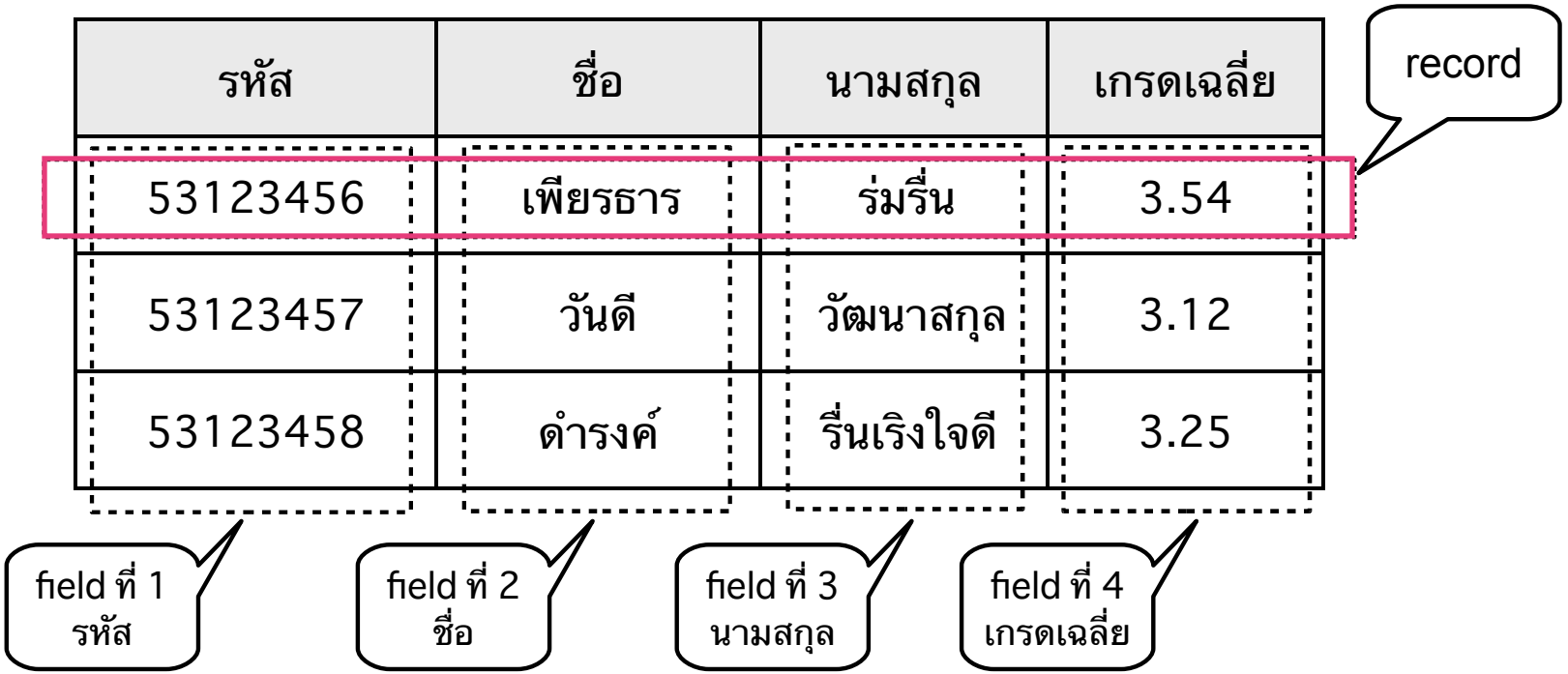
Outline

- พื้นฐานข้อมูลชนิดโครงสร้าง
- การประกาศตัวแปร
- การนิยามชนิดข้อมูล
- การกำหนดค่าเริ่มต้น
- การเข้าถึงตัวแปร

ข้อมูลชนิดโครงสร้าง

- ข้อมูลชนิดโครงสร้างเป็นการนำตัวแปรหลายตัวแปรที่มีความสัมพันธ์กันรวมเป็นโครงสร้างเดียวกัน
- ตัวแปรแต่ละตัวในโครงสร้าง มีชนิดข้อมูลที่แตกต่างกันได้

ข้อมูลชนิดโครงสร้าง



ตัวแปรชนิดโครงสร้าง

- ภาษา C ใช้ชนิดข้อมูลแบบ struct ในการเก็บข้อมูลชนิดโครงสร้าง

การประกาศตัวแปรชนิดโครงสร้าง

1. กำหนดรูปแบบของโครงสร้าง

```
struct StructName
{   // field list
    type name1;
    type name2;
    ...
};
```

การประกาศตัวแปรชนิดโครงสร้าง

2. ประกาศตัวแปร

```
struct StructName
```

ข้อมูลชนิดโครงสร้าง

รหัส	ชื่อ	นามสกุล	เกรดเฉลี่ย
53123456	เพียรธาร	ร่มรื่น	3.54
53123457	วันดี	วัฒนาสกุล	3.12
53123458	ดำรงค์	รื่นเริงใจดี	3.25

record

```
struct STUDENT
{
   char id[8];
   char first[20];
   char last[20];
   float gpa;
};
struct STUDENT stdRecord;
```

การนิยามชนิดข้อมูล (Type Definition)

- Type definition คือการนิยามชื่อของข้อมูล หรือการตั้งชื่อใหม่ให้กับชนิดข้อมูลที่มีอยู่เดิม
- เพื่อทำให้การประกาศตัวแปรง่ายและสั้น
- ใช้คำสั่ง typedef ดังนี้

```
typedef type new-type
```

ตัวอย่าง การนิยามชนิดข้อมูล

```
typedef struct
{
    char id[8];
    char first[20];
    char last[20];
    float gpa;
} STUDENT;

STUDENT stdRecord;
```

การกำหนดค่าเริ่มต้น

```
STUDENT stdRecord =
    {“1234567”,”Wandee”,”Meesuk”,3.5}
```

การเข้าถึงตัวแปร

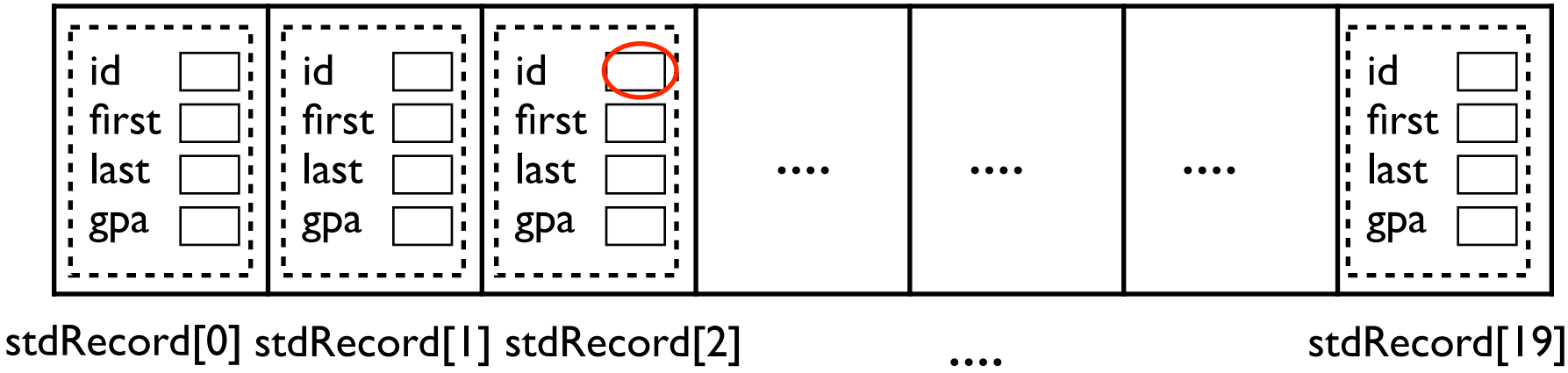
```
stdRecord.id
stdRecord.first
stdRecord.last
stdRecord.gpa
```

อาร์เรย์ของตัวแปรชนิดโครงสร้าง

```
STUDENT stdRecord[20];
```

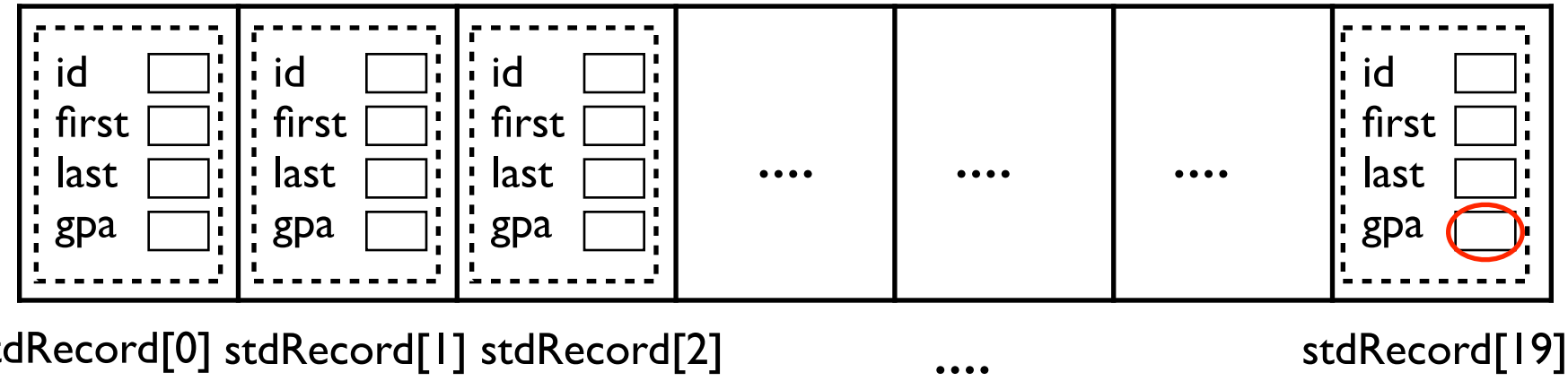
```
stdRecord[1].id = “1234567”;
stdRecord[1].gpa = 3.45;
```

อาร์เรย์ของตัวแปรชนิดโครงสร้าง




```
stdRecord[2].id
```

อาร์เรย์ของตัวแปรชนิดโครงสร้าง




```
stdRecord[19].gpa
```

โปรแกรมที่ 12.2

```
2  #define MAX_STD 20
3  typedef struct
4  {
5    char id[8];
6    char first[20];
7    char last[20];
8    float gpa;
9  } STUDENT;
10 int main()
11 {
12   STUDENT stdRecord[MAX_STD];
13   int i;
14   for (i = 0; i < MAX_STD; i++)
15   {
16      printf("\n--- Student No.%d ---", i + 1);
17      printf("\nPlease enter id : ");
18      scanf("%s", stdRecord[i].id);
19      printf("Please enter first and last name : ");
20      scanf("%s%s", stdRecord[i].first, stdRecord[i].last);
21      printf("Please enter GPA : ");
22      scanf("%f", &stdRecord[i].gpa);
23   }
24   printf("\n--- A Student Record is ---");
25   printf("\nID \t\t Name \t\t\t\t GPA\n");
26   for (i = 0; i < MAX_STD; i++)
27   {
28      printf("\n%s \t%s %s \t%15.2f", stdRecord[i].id,
29              stdRecord[i].first, stdRecord[i].last,
30              stdRecord[i].gpa);
31   }
32  return 0;
```

การใช้พอยน์เตอร์กับโครงสร้าง

```
typedef struct
{
    int a;
    float b;
    char c;
} RECORD;
```

```
RECORD rec;
RECORD* recPtr;
...
recPtr = &rec;
```

การใช้พอยน์เตอร์กับโครงสร้าง

- การเข้าถึงตัวแปร

```
(*pointer).fieldName
```

- หรือ

```
pointer->fieldName
```

Pointer to Structure

```
typedef struct
{
    int a;
    float b;
    char c;
} RECORD;
```

```
RECORD rec;
RECORD* recPtr;
...
recPtr = &rec;
```

```
(*recPtr).a
```

หรือ

```
recPtr->a
```

โปรแกรมที่ 12.3

```
4  typedef struct
5  {
6    char id[8];
7    char first[20];
8    char last[20];
9    float gpa;
10 } STUDENT;
11 int main(void)
12 {
13   STUDENT* stdPtr;
14   stdPtr = (STUDENT*) malloc (sizeof(STUDENT));
15   strcpy(stdPtr->id, "1234567");
16   strcpy(stdPtr->first, "Wandee");
17   strcpy(stdPtr->last, "Meesuk");
18   stdPtr->gpa = 3.5;
19   printf("\nID: %s ", stdPtr->id);
20   printf("\nName: %s %s", stdPtr->first, stdPtr->last);
21   printf("\nGPA: %.2f", stdPtr->gpa);
22   return 0;
}
```